Energy consumption profiling using Gaussian Processes

Christiaan Leysen*, Mathias Verbeke†, Pierre Dagnely†, Wannes Meert*
*Dept. Computer Science, KU Leuven, Belgium
†Data Innovation Team, Sirris, Belgium

***Abstract*—We present a novel clustering approach for time series based on Gaussian process regression in order to discover insights in the spending habits of households. The advantage of the proposed method is that it avoids the pairwise comparison of time series, employed by many existing methods. To this end, it learns a generalized model on several time series at once, based on their likelihood. We have validated our method using a real-world energy consumption dataset of 71 households and compared it with K-medoids and agglomerative clustering, using dynamic time warping. We not only show that our method is superior in terms of scalability but also that the produced results are useful in the decision making process of a company.**

***Keywords*—Gaussian process regression; Clustering; Time series; Energy consumption profiling**

I. Introduction

Accurate prediction of household energy consumption is of great value to energy companies as it is crucial for estimating the aggregated energy that needs to be bought by these companies. The spending behaviour of a household depends on a number of factors (e.g., family size, presence of airconditioning) and contains a lot of variation. Information about these variables leads to a better understanding of the spending habits of an energy company's clients. Often, however, these variables are not known or it is unclear what variables should be investigated and measured. It is in this context that we propose an algorithm for clustering household energy consumption data. These clusters can be used to derive spending profiles of the households or to recommend households certain energy packages. Not only the relations between households but also the relationship between consumption periods are interesting to investigate, e.g. to detect anomalies. In addition to energy companies, this information is also of interest to the households themselves, since it gives insights in their spending behaviour throughout the year.

Energy consumption is naturally represented as a time series. The households that end up in the same cluster do not necessarily have an identical consumption but their spending behaviour follows the same temporal patterns. A model of this behaviour can be learned by auto-regression methods to predict future values. In this work we will compare the learned patterns to group households together rather than to predict future values. Since we do not know upfront what time periods are relevant or how many patterns are superimposed, a method that learns this model based on the available data is preferred. For this reason and the fact that there is no need for parameter tuning by the user, we selected Gaussian Processes.

A disadvantage of GP is that it requires a single function as input while we consider a set of functions, as we want to learn a model for multiple households together. To overcome this, we present a new approach to jointly learn a GP model and optimize the hyperparameters over a set of functions, which is the first contribution of this paper. Second, we embed this in a hierarchical clustering method for time series. Finally, we present an interactive application and show how this can be used by energy providers to analyse and predict customer behaviour.

In order to test our approach we propose two experiments. In our main experiment we cluster the energy consumption of 71 distinct households to investigate which households have similar spending habits. Next we compare the time complexity of our method with K-means and agglomerative clustering, both using dynamic time warping as a similarity measure. In a second experiment, we test our method on data from four different weeks per households (and this for three households) in order to find households with steady spending habits.

II. Related work

Gaussian Processes (GPs) have been used previously for clustering. This work, however, is the first to learn over a set of functions to characterize the temporal behaviour of a cluster. Pimentel et al. [1] learn a GP only after aggregating time series. Kim et al. [2] employ a clustering method based on the variance function of Gaussian process regression in combination with a reduced complete graph strategy. However, this method clusters feature vectors instead of time series. Kumar et al. [3] propose a distance function based on the assumed independent Gaussian models of data errors and used a hierarchical clustering method to group seasonality sequences into a desirable number of clusters. The work of Duvenaud [4] presents a thorough investigation of GPs for automatically constructing, visualizing and describing a large class of models, useful for forecasting and finding structure inside time series using GPs. The clustering methods investigated in this work are also for feature vectors, not time series.

In addition to GPs, a variety of other techniques have been applied to cluster time series. Liao et al. [5] provides an overview of raw-data-based, feature-based and model-based clustering techniques. This survey concludes that until now the focus was on techniques based on statistical (e.g. ARIMA) and probabilistic methods (e.g. Dynamic Bayesian nets). This work contributes GPs, which fit within the class of model-based techniques. Mostly statistical methods have been applied on energy data [6]. The advantage of GPs as used in this work is that these do not require domain knowledge to set hyperparameters.

Rani et al. [7] provide a survey of recent techniques to cluster timer series and among others describe the use of K-means clustering with different similarity measurements. Using L*-norms with K-means is a general method but one has to consider the curse of dimensionality. Recent work of Lavin et al. [8] uses K-means with L*-norms to group and identify patterns in $k = 9$ energy profiles of a number of customers. The authors show positive results for vectors representing 24 hour periods with 15 minute intervals. When using Dynamic Time Warping (DTW) [9] as a distance metric [10], however, K-means fails to give correct results because it averages the shape of the time series that may be partially shifted [11].

Another method, named k-shape proposed by Paparrizos et al. [12] is a novel centroid-based clustering algorithm that uses the cross-correlation as similarity measurement. It outperforms all scalable and non-scalable partitional, hierarchical, and spectral methods in terms of accuracy, with as only exception K-medoids with DTW which achieves similar results. However, K-medoids needs to compute the similarity matrix, which makes it harder to scale. The best performing shape-based approaches from the literature are partitional methods combined with scale- and shift-invariant distance measures. Among partitional methods, K-medoids [13] is the most popular method as it enables the easy adoption of any shape-based distance measure.

III. Background

A. Gaussian process regression

We will first describe Gaussian processes, following the description given by Rasmussen and Williams [14]. A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution, and can be completely specified by its mean $m(x)$ and covariance function $k(x, x')$:

$$m(x) = \mathbb{E}[f(x)], \tag{1}$$

$$k(x, x') = \mathbb{E}[(f(x)m(x))(f(x')m(x'))], \tag{2}$$

The Gaussian process can then be written as

$$f(x) \sim GP(m(x), k(x, x')). \tag{3}$$

The covariance or kernel function can be seen as a similarity metric. It maps a pair of inputs (x, x') into $\mathbb{R}$. This Gaussian process model can be used for regression purposes. Suppose we have a training set $D = (x_i, y_i)|i = 1, 2, ..., n$ with observations subjected to some noise. If we want to predict a new target y_{new} given some new input data x_{new} we have to learn the underlying function which describes the training input data, assuming a Gaussian prior. The observation and the underlying function values are not identical because there is some noise ϵ. Because of this noise, we could describe the target values as follows:

$$y = f(x) + \epsilon \tag{4}$$

with the assumption that the Gaussian noise model could be described as $\epsilon \sim \mathcal{N}(0, \sigma_n^2)$.This noise assumption together with the model directly gives rise to the likelihood, i.e., the probability density of the observations given the parameters, which is factored over cases in the training set (because of the independence assumption) to give

$$p(y|X, w) = \mathcal{N}(X^T w, \sigma_n^2 I), \tag{5}$$

In Bayesian formalism there is need for a prior to express the beliefs about the parameters before the observations are observed. We put a zero mean Gaussian prior with covariance matrix Σ_p on the weights $w \sim N(0, \Sigma_p)$. Inference in the Bayesian linear model is based on the posterior distribution over the weights, computed by Bayes rule. The marginal likelihood (normalizing constant) is the integral of the likelihood times the prior. The posterior combines the likelihood and the prior, and captures everything we know about the parameters.

Specifically, we introduce the function $\phi(x)$ which maps a D-dimensional input vector x into an N dimensional feature space. Further, let the matrix $\Phi(X)$ be the aggregation of columns $\phi(x)$ for all cases in the training set and we define the covariance matrix $K = \Phi^T \Sigma_p \Phi = K(X, X)$ which also depends on the used kernel function and the σ_f^2 signal variance. Note that we shorten $\Phi(X)$ to Φ and $K(X, X)$ to K to simplify notation. Stating this, we can now outline the algorithm used for Gaussian process regression (Alg. 1).

Algorithm 1 Predictions and log marginal likelihood for Gaussian process regression.

INPUT: $X\ (inputs)$, $y\ (targets)$, $k\ (covariance\ function)$, $\sigma_n^2\ (noise\ level)$, $\sigma_f^2\ (signal\ variance)$, $x_*\ (test\ input)$, $m(X)\ (mean\ function)$

1: $L \leftarrow cholesky(K + \sigma_n^2 I)$ Calculate pos
2: $\alpha \leftarrow L^T \setminus (L \setminus y)$ Calculate pos
3: $\overline{f}_* \leftarrow m(X) + k_*^T \alpha$
4: $v \leftarrow L \setminus k_*$
5: $\mathbb{V}[f_*] \leftarrow k(x_*, x_*) - v^T v$
6: $log\ p(y|X) \leftarrow -\frac{1}{2} y^T \alpha - \sum_i logL_{ii} - \frac{n}{2} log 2\pi$

RETURN: $\overline{f}_*\ (mean), \mathbb{V}[f_*]\ (variance)$, $logp(y|X)\ (log\ marginal\ likelihood)$

The computational complexity is $n^3/6$ for the Cholesky decomposition in line 1, and $n^2/2$ for solving triangular systems in line 2 and (for each test case) in line 4 [14]. The algorithm uses the Cholesky decomposition instead of calculating the inverting the matrix directly, since it is faster and numerically more stable.

B. Hyperparameters

The covariance or kernel function typically has some parameters that need to be set, i.e., the hyperparameters of the GP. These hyperparameters are important for fitting the given data. The kernel functions used in this work are the linear kernel function and the squared-exponential kernel function. The linear kernel function has only one parameter, the signal variance σ_0^2. The squared-exponential kernel function has as parameters the signal variance σ_f^2, the noise variance σ_n^2 and the length scale l.

Algorithm 2 Overall likelihood over all time series.

INPUT: TSS $(TimeseriesSet:$ $inputs\ and\ targets\ set)$, k $(covariance\ function)$, σ_n^2 $(noise\ level)$, σ_f^2 $(signal\ variance)$

1: $likelihood_{overall} \leftarrow 0$
2: ***for*** (X, y) ***in*** TSS ***do***
3: $L \leftarrow cholesky(K + \sigma_n^2 I)$
4: $\alpha \leftarrow L^T \setminus (L \setminus y)$
5: $-\log\ p(y|X) \leftarrow \frac{1}{2} y^T \alpha + \sum_i \log L_{ii} + \frac{n}{2} \log 2\pi$
6: $likelihood_{overall} \leftarrow likelihood_{overall} + (-log\ p(y|X))$

RETURN: $likelihood_{overall}$ $(overall\ likelihood)$

IV. Method

A. Generalized model

The method we propose for clustering time series data makes use of the Gaussian process regression method. Instead of learning a model, describing only one time series, we create a model which describes a set of time series. This is a non-trivial step because the GP accepts a *single* function, whereas we want to consider *multiple* time series of the same time. In order to create a generalized model we make use of the likelihood which is calculated when we perform Gaussian process regression. We would like to learn the generalized model which maximizes the overall likelihood of the set of time series. In other words, we would like to optimize the hyperparameters of the generalized model. In order to do so, we introduce a new function (Alg. 2). Note that K is the covariance matrix, which depends on the used kernel function, the input values and the hyperparameter σ_f^2.

As generalized model, we want to find the model for which the likelihood is maximal. As Alg. 2 returns the negative likelihood, we thus need to find the hyperparameters for which the function, described in Alg. 3, is minimal. We use $TSS.X$ and $TSS.y$ to denote the set of timestamps and the set of target values of the time series set TSS respectively. By making use of the optimal hyperparameters $(\sigma_{n*}^2, \sigma_{f*}^2)$, we can calculate the optimal L and α. Using these results and the means $\overline{y}$ of the set of target values $TSS.y$, we can calculate the generalized model by making a prediction for the same period. Note that we assume that all time series X_i of the set $TSS.X$ run over the same period, so we just need to pick a random row from this set, e.g., the first one (line 3, Alg. 3).

Algorithm 3 Find generalized model

INPUT: TSS $(TimeseriesSet:$ $inputs\ and\ targets\ set)$, k $(covariance\ function)$, σ_{ni}^2 $(initial\ noise\ level)$, σ_{fi}^2 $(initial\ signal\ variance)$

1: $(\alpha, L) \leftarrow minimum(likelihood_{overall}(TSS, k, \sigma_{ni}^2, \sigma_{fi}^2))$
2: $\overline{y} \leftarrow mean(TSS.y)$
3: $x_* \leftarrow$ $firstRow(TSS.X)$
4: $\overline{f}_* \leftarrow \overline{y} + k_*^T \alpha$
5: $v \leftarrow L \setminus k_*$
6: $\mathbb{V}[f_*] \leftarrow k(x_*, x_*) - v^T v$

RETURN: $\overline{f}_*$ $(mean), \mathbb{V}[f_*]$ $(variance)$

B. Clustering

For the subsequent clustering we employ a recursive clustering approach. The method, as outlined in Alg. 5, uses the root mean square error (RMSE) between the observed time series and the generalized model. This is used as a measurement of the similarity between the time series. The clustering method makes use of three parameters: 1) a minimum cluster size (s_{min}) if we would like to have a minimum number of time series inside a cluster, 2) a similarity threshold (t_{sim}) to check if the mean similarity of the time series inside the cluster is higher or equal to the similarity threshold, and 3) a split ratio (r) used for the number of samples that need to be split into another cluster when above conditions are met (using the approach described in Alg. 4). Else we return the original cluster and check whether the newly formed cluster is different from the input list of time series. When this is not the case we return the original time series list (line 6-7, Alg. 5). Note that line 2 of Alg. 4 is a normalization step of the RMSE values of the cluster. Also note that the time series set (TSS) is reversely ordered by the corresponding RMSE values of the time series (line 4, Alg. 5). The reason we reversely order the time series set by the RMSE value is that we remove the time series with the biggest RMSE value first, based on the split ratio (r) (line 4, Alg. 4).

C. Complexity

If we assume that one time series consists of N (equidistant) samples, the learning part of the Gaussian process regression of N samples has a computational cost of $O(N^3)$, while for predicting this is $O(N^2)$. [14] When we have S time series, this needs to be repeated S times (Alg. 3). The resulting overall cost is $O(SN^3)$ plus the cost of the minimization $O(S)$. For the clustering presented here, N is fixed whereas the number of examples or series varies. Compared to pairwise approaches such as dynamic time warping, this reduces the complexity of $O(S^2)$ to $O(S)$, resulting in a better scalability. Other methods and their complexities can be found in Section V-D.

V. Experimental setup

We conducted two experiments that are useful to energy providers. In our main experiment we cluster distinct house-

Algorithm 4 Dividing a cluster

INPUT: TSS $(TimeseriesSet:$ $inputs\ and\ targets\ set)$, E_r $(list\ of\ RMSE\ values)$, t_{sim} $(similarity\ threshold)$, s_{min} $(minimum\ cluster\ size)$, r $(split\ ratio)$

1: **if** $size(TSS) > s_{min}$ **then**
2: $E_r' \leftarrow E_r / max(E_r)$
3: **if** $mean(E_r') < t_{sim}$ **then**
4: $(C_1, C_2) \leftarrow split(TSS, E_r', r)$
5: **return**(C_1, C_2)
6: **else**
7: **return** TSS
8: **else**
9: **return** TSS

RETURN: C $(Clustering)$

Algorithm 5 Clustering of time series

INPUT: TSS $(TimeseriesSet:\ inputs\ and\ targets\ set)$, k $(covariance\ function)$, σ_{ni}^2 $(noise\ level)$, σ_{fi}^2 $(signal\ variance)$, t_{sim} $(similarity\ threshold)$, s_{min} $(minimum\ cluster\ size)$, r $(split\ ratio)$

$(\overline{f}_*, \mathbb{V}[f_*]) \leftarrow findGeneralModel(TSS, k, \sigma_{ni}^2, \sigma_{fi}^2)$
for (X, y) **in** TSS **do**
  $E_r \leftarrow RMSE(y, \overline{f}_*)$
$TSS \leftarrow reverseOrderBy(TSS, E_r)$
$(C_1, C_2) \leftarrow divide(TSS, E_r, t_{sim}, s_{min}, r)$
if $C_1 == TSS$ *or* $C_2 == TSS$ **then**
  return TSS
else
  $cluster(C_1, k, \sigma_{ni}^2, \sigma_{fi}^2, t_{sim}, s_{min}, r)$
  $cluster(C_2, k, \sigma_{ni}^2, \sigma_{fi}^2, t_{sim}, s_{min}, r)$

RETURN: C $(Clustering)$

holds based on their time series from a particular week in order to investigate which households have similar spending habits. Next we compare the time complexity of our method with two other methods. In a second and smaller experiment, we will test our method on data from four different weeks per household (and this for three households) in order to find households with steady spending habits.

A. Energy Consumption Data

Our main experiment uses historical electrical consumption data of 71 distinct households. This data was provided by 3E[1], a company operating in sustainable energy consulting, research and software. The provided data, gathered in the context of the FLEXIPAC project [15], spans over one year and is comprised of the consumption of all the appliances and the heat pump of the households. A resampling to hourly intervals was applied over the time and the consumption was normalized to focus on shape and patterns.

To compare over different weeks, the timestamp was split into 'day of week' and 'hour of day'. As a convention, the week starts on Saturday to clearly visualize the weekend behaviour and to show the transition from weekend to workweek. The target variable is the aggregated consumption data. This results in $N = 168$ samples per week and $S = 71$ time series. To compare households, the same week was selected such that the weather conditions are similar. To analyse changes in usage patterns, different weeks of the same households were selected.

B. Extending pyGPs to learn over multiple functions

For the Gaussian process regression we employed the pyGPs package [16]. We extended this package to learn and optimize (hyper)parameters over multiple functions (Alg. 2) and compute predictions (Alg. 3). To find the maximized likelihood of the set of time series (Alg. 3) we relied on the L-BFGS-B algorithm implemented in SciPy [17]. As kernel function we used an addition of the squared-exponential kernel function and a linear kernel function. This combination gave the best accuracy when performing prediction. The initial hyperparameters used are: $\sigma_{f(EXP)} = \sigma_{n(EXP)} = \sigma_{LIN} = 10^{-7}$.

[1] http://www.3e.eu

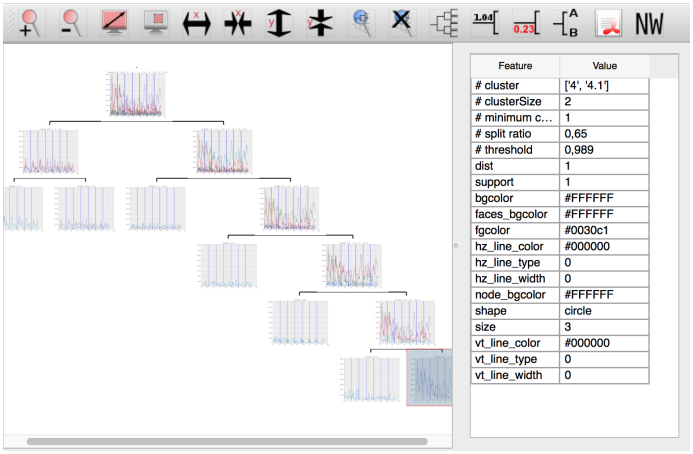

Fig. 1. Screenshot of the interactive visualization based on the ETE toolkit

C. Interactive Visualization

In order to inspect the clustering we adapted the ETE toolkit [18] to visualize the resulting clustering of the data. We converted the clustering result from Alg. 5 to the Newick tree format [19]. For each of the clusters (leafs of the tree), we create an image of the plot of the time series of the cluster. For the internal nodes, we create images of the union of the sets of time series of the underlying nodes. These images are placed in the visualization of the tree. The visualization is fully interactive and each node of the clustering is clickable and provides extra information about it (Fig. 1). This allows a user to easily inspect the resulting clustering.

D. Clustering approaches

We selected a number of clustering approaches to cluster the consumption time series of 71 distinct households.

1) Clustering using Gaussian process regression: Gaussian process regression models were learned on a set of time series. The predictive quality of the overall model with respect to each individual time series is used to select and split the set of time series to achieve a recursive clustering approach (Sec. IV). It is important to choose a good similarity threshold because when the similarity threshold is too large, the mean of the normalized RMSE values will be smaller than the similarity threshold and the clusters will continue to divide until the minimum cluster size is reached. When it is too small, the mean of the normalized RMSE values will always be greater and no new clusters will be formed because the algorithm rules that all clusters have already a high enough similarity. For the complexity experiment, we used as minimum cluster size 1 (to be able to detect unique profiles), a similarity threshold of 0.99 and a split ratio of 0.2. For the clustering of households with similar consumption we used respectively 1, 0.99, 0.25 and for checking the (in)consistent consumption behaviour we used respectively 1, 0.98 and 0.2. For the latter we used a smaller cluster similarity to slightly loosen the similarity demand because the time series are from different weeks.

Also, the split ratio is smaller to accommodate the fewer households. Note that our clustering method is deterministic, so fixed parameters will always produce the same clustering.

2) K-medoids clustering using dynamic time warping: The Partitioning Around Medoids (PAM) [13] variant of K-medoids is an existing method for clustering time series. As similarity measurement it uses DTW [9]. Its time complexity is $O(K(S-K)^2I)$ with K number of clusters, S number of time series and I number of iterations to converge [20]. To compare two temporal sequences of length N and M respectively, DTW evaluates the local cost measure for each pair of elements, resulting in a cost matrix. The time complexity of this method is $O(N^2)$ if we assume time series of the same length [10]. Calculating the similarity matrix for K-medoids using DTW, has a time complexity of $O(N^2S^2)$, thus quadratic in the number of time series. For our experiment we used the PAM algorithm described by [21] in combination with the standard DTW algorithm using the Euclidean norm as distance. A big disadvantage of this method is that the number of clusters needs to be specified a priori, which is hard, especially in our case where we do not know exactly how many clusters are desirable for getting an insight in the households. We used $K = 15$ (and $K = 5$ for the test of 10 households) based on several results of our own method. Note that we did not use the K-means algorithm because it is discouraged to use this algorithm in combination with DTW (Sec. II).

3) Hierarchical clustering using dynamic time warping: We use bottom-up hierarchical clustering or agglomerative clustering using single-linkage with a complexity of $O(S^2)$ where S is the number of time series that need to be clustered [22], [23]. The calculation of the distance matrix, with complexity $O(N^2)$, can be done separately so the overall complexity is $O(S^2N^2)$.

VI. Results

A. Households with similar consumption behaviour

We select the same week for every household and apply the selected clustering algorithms. The same week is selected because the geographical distribution allows us to assume that these households experience similar weather conditions. When applying GP clustering we obtain the results shown in Fig. 2. From Fig. 2, it can be seen that the left branch contains households with different spending habits during the weekend (the weeks start at Saturday in this plot). In the right branch and the branches beneath the first left node, we find that households with the same day/night habits are clustered together. The result of our method was validated by a domain expert. We could not compare its quality with the clusterings of the other methods due to the lack of a suitable measure. A hierarchical clustering is preferred because it gives a more structured view on what households are (dis)similar. The visualization allows to interpret distance in the tree as a proxy for the distance between two spending profiles. An advantage of GP clustering is the beneficial time complexity. Compared to K-medoids DTW and agglomerative DTW clustering, GP is linear instead of quadratic in the number of households because it employs a model based approach (Fig. 3). For datasets less than about 50 time series K-medoids and agglomerative clustering will be faster because the model learning part of our approach takes more time than the calculation of the DTW similarity for the other methods. If more than 50 time series need to be clustered, our approach is faster, because we do not need to do a pairwise comparison of all the time series.

B. Households with (in)consistent consumption behavior

For our second experiment we investigate the spending habits of households during the year. This is achieved by selecting multiple weeks of the same household. For ease of presentation, we only show the clustering for three distinct households and one week in every season (Fig 4) but one can add arbitrarily many households. By inspecting the distribution of a household over different clusters, it is clear that all weeks of household 2 are in the same cluster, and the weeks of household 3 and 1 are spread across two and four clusters respectively. This means that household 2 is the household with most consistent spending habits and household 1 is inconsistent. A visual check of the time series using our interactive tool confirms this. The time series of household X are named $X.1$, $X.2$, $X.3$, and $X.4$ for a Winter, Spring, Summer and Autumn week respectively. We can also conclude that household 3 has similar Autumn and Winter spending habits because they are clustered together. This is also true for the Spring and Summer spending habits.

VII. Practical use in decision support

The proposed method can support companies operating in the energy sector, e.g., 3E, in tackling a number of common tasks more efficiently. The first experiment showed how to cluster households in function of the occupancy (intermittent versus permanent). This allows one to benchmark the electricity consumption per cluster, and only compare households to peers which have a similar consumption behaviour. Another use of the first experiment is to identify inefficient or problematic heating systems. For example, a heat pump with repetitive large peaks in consumption is bad for energy performance, heat pump life time and the electricity grid. Such inefficiencies can be detected by looking for households that are alone in a cluster. The second experiment shows how to detect if the household exhibits a smooth and stable behaviour over time. For such a household, forecasting techniques to predict the consumption day-ahead are more accurate. Furthermore, it can be useful information when recommending certain energy

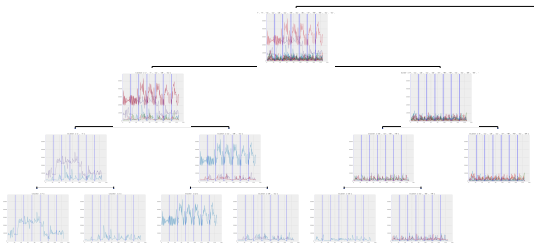

Fig. 2. Cutout of the GPR clustering (full version in appendix).

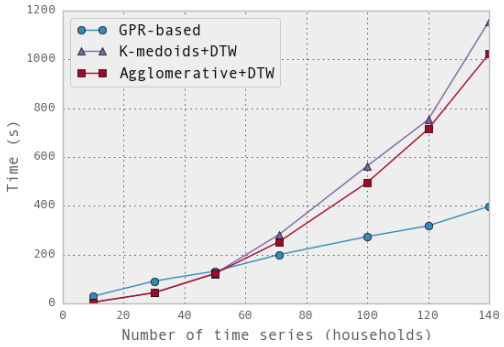

Fig. 3. Time complexity comparison of the different clustering methods. Multiple weeks were used to obtain up to 140 time series.

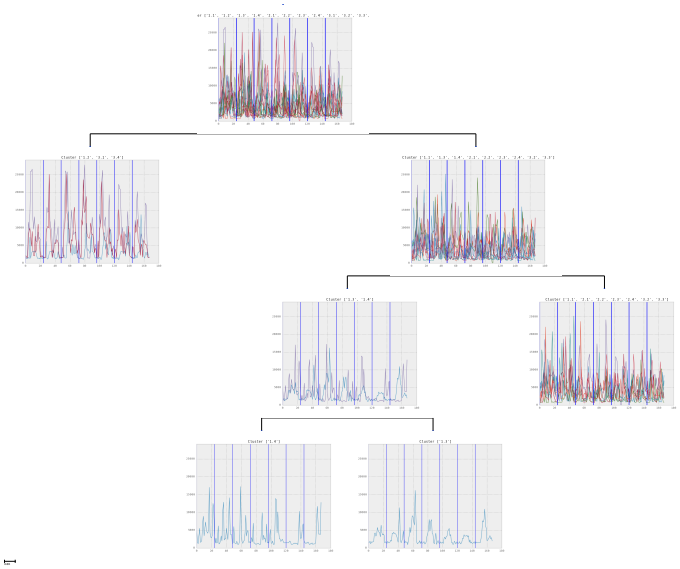

Fig. 4. Resulted clustering of three households using data from four different times. Note that we use $X.1, X.2, X.3, X.4$ to indicate the time series of household number X.

packages. In addition, our method is scalable and it does not require any parameter tuning by the user to create a model. This supports its practical usefulness.

VIII. Conclusion

We proposed an algorithm for clustering energy consumption of households. These clusters can be used to create spending profiles and give companies specific insights in the spending behaviour. In our first experiment we clustered 71 households using our Gaussian process regression-based method. First we proved that our algorithm has a linear time complexity in comparison to the quadratic time complexity of K-medoids and agglomerative clustering methods using dynamic time warping and discussed our results. In a second experiment we showed that clustering time series data from different weeks per household makes it possible to cluster households based on their steady spending habits during the year. Lastly we showed the practical use of our resulted clustering in the decision process of a company operating in the energy sector.

Acknowledgments

The authors would like to thank Clara Verhelst (iLab, 3E), Marion Neumann, Roman Garnett, Luc De Raedt, Tom Tourwé for their feedback.

References

[1] M. A. Pimentel, D. A. Clifton, and L. Tarassenko, "Gaussian process clustering for the functional characterisation of vital-sign trajectories," in *Machine Learning for Signal Processing, 2013 IEEE International Workshop on*, 2013, pp. 1–6.

[2] H.-C. Kim and J. Lee, "Clustering based on Gaussian processes," *Neural computation*, vol. 19, no. 11, pp. 3088–3107, 2007.

[3] M. Kumar, N. R. Patel, and J. Woo, "Clustering seasonality patterns in the presence of errors," in *Proceedings of the Eighth ACM International Conference on Knowledge Discovery and Data Mining*, NY, USA, 2002, pp. 557–563.

[4] D. Duvenaud, "Automatic model construction with Gaussian processes," Ph.D. dissertation, Computational and biological learning laboratory, University of Cambridge, 2014.

[5] T. Warren Liao, "Clustering of time series data - a survey," *Pattern Recogn.*, vol. 38, no. 11, pp. 1857–1874, Nov. 2005.

[6] M. Espinoza, C. Joye, R. Belmans, and B. D. Moor, "Short-term load forecasting, profile identification, and customer segmentation: a methodology based on periodic time series," *Power Systems, IEEE Transactions on*, vol. 20, no. 3, pp. 1622–1630, 2005.

[7] S. Rani and G. Sikka, "Article: Recent techniques of clustering of time series data: A survey," *International Journal of Computer Applications*, vol. 52, no. 15, pp. 1–9, August 2012.

[8] A. Lavin and D. Klabjan, "Clustering time-series energy data from smart meters," *Energy Efficiency*, vol. 8, no. 4, pp. 681–689, 2014.

[9] C. S. Myers, "A comparative study of several dynamic time warping algorithms for speech recognition," Ph.D. dissertation, MIT, 1980.

[10] H. Izakian, W. Pedrycz, and I. Jamal, "Fuzzy clustering of time series data using dynamic time warping distance," *Engineering Applications of Artificial Intelligence*, vol. 39, pp. 235–244, 2015.

[11] V. Niennattrakul and C. A. Ratanamahatana, "On clustering multimedia time series data using k-means and dynamic time warping," in *2007 International Conference on Multimedia and Ubiquitous Engineering*, April 2007, pp. 733–738.

[12] J. Paparrizos and L. Gravano, "k-Shape: Efficient and Accurate Clustering of Time Series," in *Proceedings of the 2015 ACM SIGMOD*, NY, USA, 2015, pp. 1855–1870.

[13] L. Kaufman and P. J. Rousseeuw, *Finding groups in data : an introduction to cluster analysis*, ser. Wiley series in probability and mathematical statistics. New York: Wiley, 1990.

[14] C. E. Rasmussen and C. K. I. Williams, "Gaussian processes for machine learning," 2006.

[15] "FLEXIPAC: Valorisation de la flexibilité des pompes à chaleurs," Research project financed by the Walloon Region, Belgium, 2013-2015. [Online]. Available: http://www.flexipac.ulg.ac.be

[16] M. Neumann, S. Huang, D. E. Marthaler, and K. Kersting, "pyGPs–a python library for Gaussian process regression and classification," *Journal of Machine Learning Research*, vol. 16, pp. 2611–2616, 2015.

[17] E. Jones, T. Oliphant, P. Peterson *et al.*, "SciPy: Open source scientific tools for Python," 2001. [Online]. Available: http://www.scipy.org/

[18] J. Huerta-Cepas, F. Serra, and P. Bork, "Ete 3: Reconstruction, analysis, and visualization of phylogenomic data," *Molecular biology and evolution*, p. msw046, 2016.

[19] D. R. Heckendorn. (2012) Newick tree formats. [Online]. Available: http://marvin.cs.uidaho.edu/Teaching/CS515/newickFormat.html

[20] G. W. Claude Sammut, *Encyclopedia of Machine Learning*. Springer-Verlag,N.Y., 2007.

[21] Q. Zhang and I. Couloigner, *Computational science and its applications – International Conference, Singapore, May, Part III*. Springer Berlin Heidelberg, 2005, pp. 181–189.

[22] C. D. Manning, P. Raghavan, H. Schütze *et al.*, *Introduction to information retrieval*. Cambridge university press Cambridge, 2008, vol. 1.

[23] B. Everitt, S. Landau, and M. Leese, "Cluster analysis." *Arnold, London*, 2001.

APPENDIX A
CLUSTER RESULT

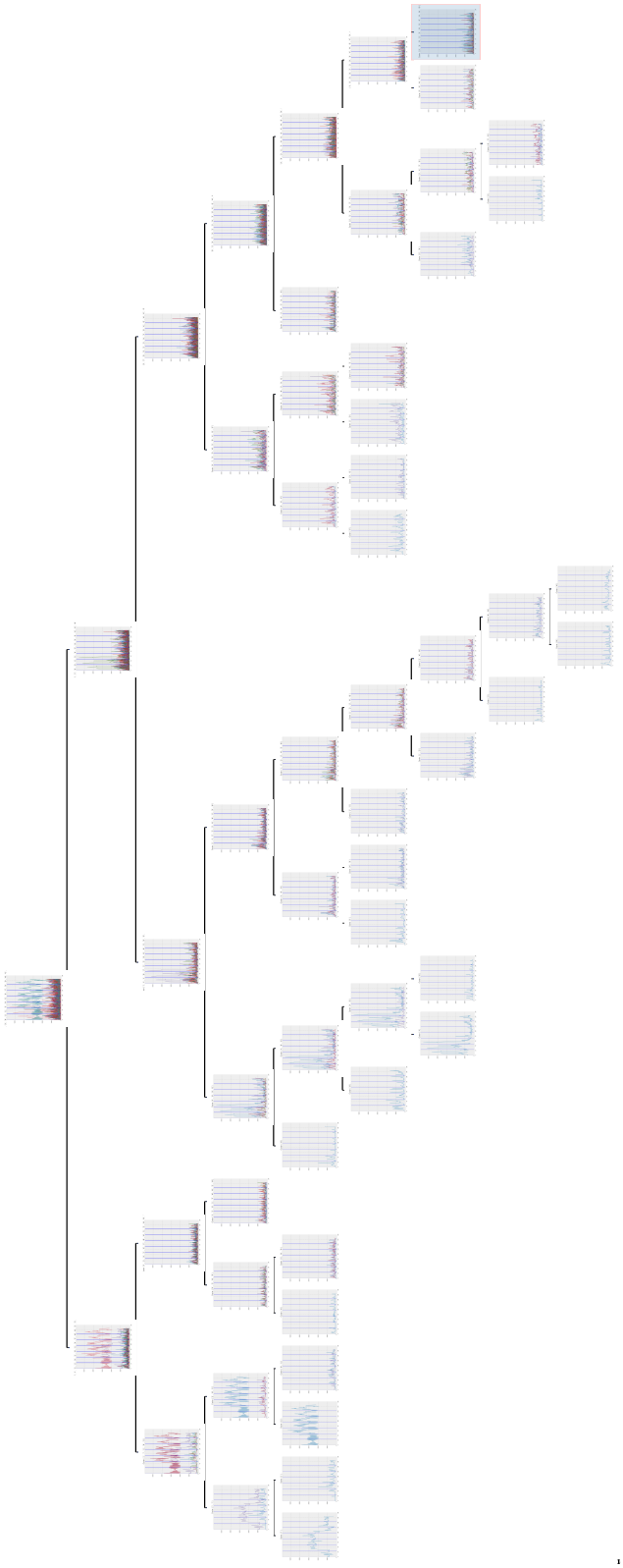

Fig. 5. Resulted clustering of our Gaussian process regression based method.